REMARKS

Claims 1-16 are all the claims presently pending in the application. Claims 1-2 and 10 are amended to more clearly define the invention. Claims 1, 10, and 14 are independent.

These amendments are made only to more particularly point out the invention for the Examiner and not for narrowing the scope of the claims or for any reason related to a statutory requirement for patentability.

Applicant also notes that, notwithstanding any claim amendments herein or later during prosecution, Applicant's intent is to encompass equivalents of all claim elements.

Claims 1-16 stand rejected under 35 U.S.C. § 102(b) as being anticipated by the Maeda reference. Claims 1-16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the Imanishi et al. reference in view of the Dessloch et al. reference.

These rejections are respectfully traversed in the following discussion.

I. THE CLAIMED INVENTION

An exemplary embodiment of the present invention, as defined by, for example, independent claim 1 is directed to a task system that includes a storage for storing one or more event identifiers for an event of a plurality of events, a task control device for creating a task based on the event, and a task processing device for executing a plurality of tasks. Whereupon completing a first task of the plurality of tasks, the task processing device initiates a search for another event identifier, and if the another event identifier is the same as the one or more event identifiers corresponding to the first task, then processes a second task, which has an event identifier that is the same as the first task, using a resource used by the first task.

Conventional task processing systems have problems improving the speed of the task processing because it has been impossible to schedule the tasks beforehand and it has been necessary to generate overhead for acquiring and releasing the resources that are necessary for task processing.

Further, conventional task processing systems are not suitable for processing a large number of small tasks, because a lot of overhead for task switching has been necessary.

In stark contrast, an exemplary embodiment provides a task system that stores event identifiers and that, upon completion of a first task, searches those event identifiers to process a second task that has an event identifier that is the same as the first task and which uses the same resource that was used by the first task. In this manner, overhead for acquiring and releasing resources is reduced which increases the speed of task processing. (Page 3, lines 3-8). For example, it becomes possible to process a large number of small tasks at a higher speed than has conventionally been possible. (Page 3, lines 13-15).

This feature of an exemplary embodiment of the invention consists in that after completing a first task, a task processing device searches whether or not an event identifier of an event corresponding to the completed task is registered in a storage, and if it is registered, the task processing device processes a second task corresponding to the event registered event identifier using a resource used by the first task, thereby reducing the overhead for acquiring and releasing resources that are necessary for processing the same type of task.

II. THE PRIOR ART REJECTIONS

A. The Maeda reference

Regarding the rejection of claims 1-16, the Examiner alleges that the Maeda reference

teaches the claimed invention. Applicant submits, however, that there are elements of the claimed invention which are neither taught nor suggested by the Maeda reference.

The Maeda reference does not teach or suggest the features of the present invention including a storage for storing one or more event identifiers for an event of a plurality of events and a task processor that both initiates a search for another event identifier and that processes a second task if the another event identifier is the same as the event identifier for the first task, where the second task uses the same resource that was used by the first task.

Rather, and in stark contrast, the Maeda reference is only concerned with controlling the number of tasks that are simultaneously executed in order to control the load balance between task groups.

The Maeda reference explains that there are two different types of tasks. There are proper tasks, which may only be performed using a specific resource, and a share task, which can access all resources ([0007]). The Maeda reference further describes a job 101 that includes an event storing queue 107 that receives events from other jobs ([0008]), an originating task 104 that includes a task management means 109, a tasks management table 108 and grouping sub-tasks into proper tasks 1, proper tasks 2, and share tasks 106. The proper tasks 1 and proper tasks 2 are grouped into different groups where each group is related to a specific resource that is only capable of performing the proper task within each group. Of course, the share tasks 106 may access any resource. ([0008], [0010], and [0018]).

The Maeda reference explains that the task management means 109 assigns a sub-task from an event taken from either the event storing queue 107 or which is just received at 102, based upon an analysis of the tasks managed table 108. ([0018]).

As illustrated in Figure 3, the tasks managed table 108 includes information regarding

the number of shared tasks that the system is capable of concurrently processing at 36 and the total number of shared tasks that are currently being processed at 37. ([0021]).

The tasks managed table 108 also stores a table that is organized based upon a group identifier 31. Each row within the tasks managed table 108 is specific to a type of resource identified by the group identifier 31. For example, Figure 6 has two rows. The first row is identified by a group identifier "A," which is specific to a first resource, while the second row is identified by a group identifier "B," which is specific to a second resource that is different from the first resource. In other words, the group identifier 31 is a resource identifier, not a task identifier or an event identifier.

Further, within each row, the tasks managed table 108 includes the number of proper tasks that are useable by that resource 32, the number of share tasks that are useable by that resource 33, the number of proper tasks that are currently being processed by that resource 34, and the number of shared tasks that are currently being processed by that resource 35. ([0021] - [0022]).

As each event is received, the task management means 109 classifies the event according to the type of resource (i.e., as it corresponds to a resource/group identifier 31) and registers the event into the event storing queue 107. ([0024]).

The tasks managed table 108 is updated as events are received by the task management means 109. If an event 103 is received by the task management means 109 from one of the proper task groups (1 or 2 in Figure 1) or from the shared task group 106, then the task management means 109 reduces the corresponding number within the tasks managed table. ([0025] - [0027]). For example, if the event 103 includes a shared task, then the task management means 109 reduces the total number of shared tasks being processed

indicator at 37 by one and also reduces the number of shared tasks currently being processed from column 35 within the corresponding group (i.e. resource row). The task management means 109 performs a similar reduction with respect to the number of proper tasks being currently processed at column 34 for a proper task.

Since, a task has just been completed, obviously, there are resources available for processing further tasks. Thus, the Maeda reference explains that the another event is selected for processing.

In particular, the Maeda reference explains that the task management means 109 chooses one event 102 from the event storing queue 107 at step 403. However, the Maeda reference does not specify how the next event 102 is selected from the event storing queue 107.

Indeed, the Maeda reference makes it clear that it does not matter how the next event 102 is selected from the event storing queue and states that the selection may be made according to a FIFO, round robin, etc. and that "the selection approach of the event 102 from this event storing queue 107 may use any approach in this invention, and does not specify it in this example." ([0029]).

The Maeda reference also explains that if there is no event in the queue 107, that the task management means 109 simply waits until another event 102 is received. ([0030]).

Therefore, the Maeda reference clearly explains that the events are chosen simply on the basis of their presence within the event storing queue 107 and that it does not matter how the next event is selected for processing. ([0029]).

Clearly, the Maeda reference does not teach or suggest searching event identifiers to process a second task that has an event identifier that is the same as the first task where the

second task uses the same resource that was used by the first task..

Rather, the Maeda reference teaches that it does not matter which event is selected from the event storing queue 107.

Indeed, the Maeda reference does not teach or suggest performing any search at all, let alone searching event identifiers to process a second task that has an event identifier that is the same as the first task, or where the second task uses the same resource that was used by the first task.

Moreover, the Maeda reference clearly suffers from the same problems that are solved by the present invention. The system that is disclosed by the Maeda reference does not teach or suggest anything at all that reduces the amount of resource allocating and releasing and, therefore, suffers from exactly the same problems that are solved by the present invention.

Since the Maeda reference does not teach or suggest searching event identifiers to process a second task that has an event identifier that is the same as the first task, the Maeda reference is not capable of reducing the number of times that overhead is acquired and released in order to increase the speed of task processing. So that, for example, it becomes possible to process a large number of small tasks at a higher speed than has conventionally been possible as the present invention is capable of providing.

Rather, the Maeda reference is only concerned with controlling the load balance between groups of a task ([0042])

Therefore, the Maeda reference does not teach or suggest each and every element of the claimed invention and the Examiner is respectfully requested to withdraw this rejection of claims 1-16.

B. The Imanashi et al. reference in view of the Dessloch et al. reference

The Examiner alleges that the Dessloch et al. reference would have been combined with the Imanashi et al. reference to form the claimed invention. Applicant submits, however, that these references would not have been combined and even if combined, the combination would not teach or suggest each and every element of the claimed invention.

Neither of the Imanashi et al. reference nor the Dessloch et al. reference teaches or suggests event identifiers at all, let alone conducting a search of event identifiers, or searching event identifiers to process a second task that has an event identifier that is the same as the first task, or where the second task uses the same resource that was used by the first task.

Rather, the Imanashi et al. reference discloses a task switching system that controls task switching based upon an identification of the hardware engine that is required to execute the task. The Imanashi et al. reference relies upon a hardware scheduler that reflects an execution priority of the tasks, which makes it possible to select a task to be run next without re-determining which of the hardware engines executes a time critical process at the time of task switching.

In particular, the Imanashi et al. reference discloses a terminated core determination unit (TCDU) 332 that receives a termination signal 124 sent from a core 111-115 and that then identifies a task allocated to that execution-terminated core. "Such identification is carried out with reference to the task management table 310 and a task number 362." (Col. 5, lines 5-10).

Contrary to Examiner's allegation, the Imanashi et al. reference does not teach or suggest any event identifier at all, let alone storing event identifiers.

The Examiner cites column 5, lines 1-20 of the Imanashi et al. reference in an attempt

to support the allegation. However, column 5, lines 1-20 only makes reference to a task identifier and does not mention anything at all regarding an event identifier.

The Dessloch et al. reference does not remedy the deficiencies of the Imanashi et al. reference.

Indeed, the Dessloch et al. reference is non-analogous art.

> "In order to rely on a reference as a basis for rejection of an applicant's invention, the reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned." (M.P.E.P. § 2141.01(a))

In this instance, the Dessloch et al. reference is neither 1) within the field of applicant's endeavor; or 2) reasonably pertinent to the particular problem with which the inventor was concerned.

The Dessloch et al. reference is not within the field of applicant's endeavor.

The field of the applicant's endeavor is the field of task processing. (Page 1, lines 6 - 10).

In stark contrast, the Dessloch et al. reference is in the field of database management systems (col. 1, lines 63-67).

Indeed, the Dessloch et al. reference does not teach or suggest anything at all that is even remotely related to task processing.

Clearly, the Dessloch et al. reference is not within the field of applicant's endeavor.

The Dessloch et al. reference is also not reasonably pertinent to the particular problem with which the inventor was concerned.

As explained above, the inventor was concerned with the particular problem of

improving the speed of task processing as it relates to the release and acquisition of resources, especially with respect to increasing the processing a large number of small tasks (page 2, lines 5 - 14).

In stark contrast, the Dessloch et al. reference is concerned with the completely different and unrelated problem of supporting the management of relational databases such that they may efficiently handle new data types and specifically toward providing a new approach to the support of indexing of semi-structured data in a relational database system. (Col. 2, line 31 - col. 3, line 23).

Clearly, the Dessloch et al. reference has absolutely nothing to do with the problem of improving the speed of task processing.

Indeed, the Dessloch et al. reference does not mention anything at all that is even remotely related to task processing.

The Dessloch et al. reference is clearly not reasonably pertinent to the particular problem with which the inventor was concerned.

Therefore, the Dessloch et al. reference is clearly non-analogous art and, as a result, the Examiner is not allowed "to rely on" the Dessloch et al. reference "as a basis for rejection of an applicant's invention" as is clearly set forth by M.P.E.P. § 2141.01(a).

Further, not only is the Dessloch et al. reference not available for use against the present application for being non-analogous, but Applicant submits that these references would not have been combined as alleged by the Examiner. Indeed, the references are directed to completely different matters and problems.

Specifically, the Imanashi et al. reference is directed to the problem of switching between tasks at a high speed (col. 1, lines 32-35).

In stark contrast, as explained above, the Dessloch et al. reference is directed to the completely different and unrelated problem of supporting the management of relational databases such that they may efficiently handle new data types and specifically toward providing a new approach to the support of indexing of semi-structured data in a relational database system.

One of ordinary skill in the art who was concerned with the problem of switching between tasks at a high speed as the Imanashi et al. reference is concerned with solving would not have referred to the Dessloch et al. reference because the Dessloch et al. reference is only concerned with the completely different and unrelated problem of supporting the management of relational databases such that they may efficiently handle new data types and specifically toward providing a new approach to the support of indexing of semi-structured data in a relational database system. Thus, the references would not have been combined.

Therefore, the Examiner is respectfully requested to withdraw the rejection of claims 1-16.

III. FORMAL MATTERS AND CONCLUSION

In view of the foregoing amendments and remarks, Applicant respectfully submits that claims 1-16, all the claims presently pending in the Application, are patentably distinct over the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the Application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

The Commissioner is hereby authorized to charge any deficiency in fees or to credit any overpayment in fees to Attorney's Deposit Account No. 50-0481.

Respectfully Submitted,

Date: 1/31/06

James E. Howard
Registration No. 39,715

McGinn Intellectual Property Law Group, PLLC
8321 Old Courthouse Rd., Suite 200
Vienna, Virginia 22182
(703) 761-4100
Customer No. 21254